Efficient LoRA-Based Guardrail Models for Safe LLM Deployment on Consumer GPUs

Mulla Zaheer Ahamed
Research & Development
Samsung R&D Institute - India
Bengaluru, India
mulla.zaheer@partner.samsung.com

Ravindra SR
Research & Development
Samsung R&D Institute - India
Bengaluru, India
ravindra.sr@samsung.com

Dr. Uma Devi M
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
umadevim@srmist.edu.in

Dr. Saranya S S
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
saranyas6@srmist.edu.in

Shashvat Aghera
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
sa8055@srmist.edu.in

Suhas Thammysetty
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
st2721@srmist.edu.in

Sanchi Manchanda
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
sm8280@srmist.edu.in

Anisha Rakshit
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
ar8278@srmist.edu.in

***Abstract*—Although Large Language Models (LLMs) demonstrate strong generative capabilities, their deployment introduces significant safety risks. These risks include generating harmful, unsafe, or policy-violating content. Guardrail systems are essential for moderating both user inputs and model outputs to ensure safe and responsible interactions. This work explores effective guardrail methods for safety classification in five categories: safe, toxic_content, malicious_code, harmful_instruction, and self_harm. In the first phase, we assess prompt-engineered guardrails across various open-source LLMs using a curated part of the NVIDIA Aegis 2.0 safety dataset. Models like Gemma3-4B show impressive classification performance, reaching about 97% accuracy in input safety detection. However, these methods struggle with adversarial prompts and complex context changes. To solve these problems, we create a dedicated safety classifier by fine-tuning the Falcon-1B model with Low-Rank Adaptation (LoRA). The model is trained on around 12,000 prompts, including 500 adversarial jailbreak examples, using an 80-10-10 split for training, validation, and testing. The new model achieves a class accuracy of 0.984 and a macro F1-score of 0.982. It outperforms several guardrail baselines, such as Granite Guardian and NVIDIA Nemotron models. Additionally, the system has low leakage (FNR = 0.0157) and provides real-time inference with an average latency of 131.9 ms per prompt on a consumer-grade RTX 4060 GPU. These results show that small, specialized guardrail models can do better than larger moderation systems while being efficient enough for practical use.**

I. Introduction

Large Language Models (LLMs) have become widely used for generating coherent and contextually relevant text. These models are increasingly deployed in applications such as chatbots, content generation systems, educational tools, healthcare assistance, and financial services. As their adoption continues to grow, ensuring the safety and reliability of their outputs has become an important concern.

When LLMs are used in real-world settings, they may occasionally generate responses that violate safety guidelines. For example, a model might produce toxic language, provide harmful advice, generate malicious code, or discuss self-harm in an unsafe manner. In sensitive domains, such responses can lead to serious consequences, making it essential to incorporate mechanisms that prevent unsafe outputs.

To address these concerns, safety mechanisms known as *guardrails* are commonly used. Guardrails act as monitoring layers that analyze both user inputs and model-generated responses. They can intervene at multiple stages of the interaction pipeline, such as when a user submits a prompt, while the model is generating a response, or before the response is delivered to the user. In many implementations, guardrails function as filtering systems that classify prompts or responses into categories such as safe or unsafe.

One common approach for improving safety in LLM systems is instruction-based prompting, where models are provided with explicit safety instructions and example interactions. While this method can help guide model behavior, it does not always perform reliably. Users may attempt to bypass safeguards through adversarial prompts, and ambiguous contexts can lead to incorrect responses. Furthermore, many existing safety models are large and computationally expensive, which limits their practicality for real-time deployment.

In this work, we propose a lightweight guardrail model

designed to classify prompts into safety-related categories, including safe content, toxic content, malicious code, harmful instructions, and self-harm. We first evaluate instruction-based prompting using existing LLMs as a baseline approach. We then develop a dedicated safety filter by fine-tuning the Falcon-1B model using a dataset of approximately 12,000 examples, including adversarial prompts intended to challenge the system.

The main contributions of this work are as follows:

- Evaluation of instruction-based prompting for safety classification using existing LLMs.
- Development of a lightweight safety filter through fine-tuning the Falcon-1B model.
- Creation of a dataset containing approximately 12,000 examples, including adversarial prompts.
- Demonstration that the proposed safety filter can operate in real time on standard computing hardware.

The experimental results indicate that specialized safety filters can effectively improve the reliability of LLM-based systems while remaining computationally efficient for practical deployment.

II. Literature Review

This section reviews prior research in three areas relevant to this work: (1) responsible AI and safety mechanisms for large language models (LLMs), (2) LLM deployment in sensitive domains requiring strict moderation, and (3) production-oriented guardrail and deployment frameworks for LLM services.

Research on responsible AI and LLM safety has grown rapidly, focusing on detecting and mitigating harmful model behavior. Recent studies highlight the effectiveness of alignment techniques such as chain-of-thought (CoT) alignment and explanation-aware fine-tuning for improving the reliability of LLM-based safety evaluators. These approaches enable models to act as "LLM-as-a-judge" systems capable of identifying unsafe content even with limited labeled data while revealing trade-offs between reasoning transparency and prediction reliability [1].

Other work explores prompt-constrained classifiers and structured output prompting strategies, comparing them with supervised and instruction-tuned moderation models. These studies propose structured templates and synthetic data generation methods to improve the robustness and consistency of safety classifiers [3], [4]. Evaluation methodologies have also evolved to include multi-metric benchmarking and paraphrase robustness testing, highlighting ongoing challenges such as ambiguity between closely related categories (e.g., malicious_code vs. harmful_instruction) [3], [6].

The deployment of LLMs in sensitive domains such as education further emphasizes the need for strong safety guarantees. Prior work proposes "ethics-by-design" frameworks that integrate policy engines, rule-based constraints, and AI-assisted moderation tools to ensure compliance with institutional policies [5]. These systems are particularly important in K–12 environments where content appropriateness and reliability are critical. Such systems often combine rule-based filtering with learned classifiers to enforce safety policies in real-time environments [7], [8].

From a systems perspective, lightweight fine-tuning techniques such as adapter-based training and Low-Rank Adaptation (LoRA) have emerged as efficient methods for adapting large language models without extensive retraining [1], [3]. These approaches allow targeted specialization of models while maintaining computational efficiency, making them well suited for real-time moderation systems operating on limited hardware. Recent studies suggest that combining deterministic safety rules with lightweight neural classifiers can form effective hybrid guardrail architectures where rule-based filters enforce high-confidence policies and fine-tuned models handle nuanced classification tasks.

Recent work by Kazemi Rad et al. (2025) further investigates guardrail systems for conversational AI, demonstrating that targeted alignment and fine-tuning strategies can significantly improve adversarial prompt detection. Their study explores several modern LLM architectures and proposes training strategies based on preference alignment methods such as Direct Preference Optimization and Kahneman–Tversky Optimization. These approaches improve moderation accuracy and the interpretability of safety decisions, enabling guardrail models to outperform existing moderation systems such as LlamaGuard-2, DeBERTaV3, and PromptGuard while reducing false alarms and improving robustness to adversarial prompts.

Overall, these studies emphasize the increasing need for scalable and efficient safety classifiers for LLM-based systems. They indicate that lightweight fine-tuning strategies, when combined with structured safety datasets and adversarial evaluation, can provide effective and practical guardrail solutions for real-world AI deployments.

TABLE I: Summary of Prior Work in LLM Safety and Guardrails

Work	**Focus**	**Limitation**
Hu et al. (2022) [1]	LoRA fine-tuning	Not safety-focused
Chen et al. (2023) [3]	Efficient LoRA variants	No moderation task
Qi et al. (2025) [4]	LLM safety analysis	No guardrail model
Akiri et al. (2025) [5]	Risk assessment	No deployment method
He et al. (2023) [8]	Code security risks	Code domain only
Song et al. (2021) [7]	Toxic text detection	Not LLM-specific
This Work	LoRA Falcon guardrail	Real-time classifier

III. Methodology

This section describes the design of the proposed guardrail moderation system, including the problem formulation, dataset construction, baseline evaluation, and the LoRA-based fine-tuned safety classifier.

A. Problem Definition

In real-world conversational systems, user prompts may contain harmful or policy-violating content such as toxic language, malicious code requests, harmful instructions, or self-harm related queries. To mitigate such risks, guardrail systems must detect unsafe interactions before they reach the downstream language model.

We formulate guardrail moderation as a multi-class safety classification task. Given an input prompt x, the classifier predicts a safety label y:

$$y = f_\theta(x)$$

where f_θ represents the safety classification model and y denotes one of the predefined safety categories.

The classification schema used in this work consists of five categories:

- safe — none
- unsafe — toxic_content
- unsafe — malicious_code
- unsafe — harmful_instruction
- unsafe — self_harm

During inference, prompts classified as unsafe are blocked or flagged by the guardrail layer, while safe prompts are forwarded to the downstream LLM for response generation.

B. Dataset Construction

Two datasets were used during the development of the guardrail system.

Phase 1: Prompt Engineering Evaluation

For initial experiments, a curated subset of approximately 460 prompts was extracted from the NVIDIA Aegis safety dataset. These prompts were balanced across four unsafe categories (toxic content, harmful instructions, self-harm, and malicious code), along with 50 safe prompts used as control samples. This subset was used to evaluate prompt-engineered guardrails across several open-source LLMs.

Phase 2: Fine-Tuning Dataset

For training the dedicated safety classifier, a larger dataset of approximately 12,000 prompts was constructed. The dataset includes:

- safety prompts derived from the Aegis dataset
- additional adversarial examples
- 500 jailbreak prompts simulating attempts to bypass guardrails

Each data instance contains the following fields:

- Prompt: input text
- Label: safe or unsafe
- Threat: specific unsafe category

All samples were normalized into the final classification schema used by the guardrail model.

C. Dataset Split

To maintain balanced class distributions, the dataset was divided using stratified sampling:

- Training set: 80%
- Validation set: 10%
- Test set: 10%

Additional dataset validation steps were performed, including inspection of class distributions, token length analysis, and anomaly checks to ensure dataset quality.

D. Baseline Guardrail Evaluation

Before training the dedicated safety classifier, prompt-engineered guardrails were evaluated using several open-source LLMs:

- Gemma3-4B
- Qwen-2.5-7B
- Mistral-Instruct
- Llama-Guard-3-8B

While prompt engineering achieved reasonable performance in controlled settings, the models exhibited limitations when handling adversarial prompts and ambiguous safety categories.

E. LoRA-Based Guardrail Classifier

To improve classification stability and efficiency, a dedicated safety classifier was developed by fine-tuning the Falcon-1B transformer model.

Falcon-1B provides a suitable trade-off between model capacity and computational cost, enabling real-time inference on consumer-grade hardware.

Parameter-efficient fine-tuning was performed using Low-Rank Adaptation (LoRA), which introduces trainable rank-decomposition matrices into the model layers while keeping the original weights frozen. This significantly reduces the number of trainable parameters.

LoRA adapters were applied to the following layers:

- Attention projection layers
- MLP projection layers

TABLE II: LoRA Configuration

Parameter	Value
Rank	8
Alpha	16
Dropout	0.05

All prompts were tokenized using the Falcon tokenizer and truncated or padded to a maximum sequence length of 256 tokens.

TABLE III: Training Configuration

Hyperparameter	Value
Batch size	4
Learning rate	$2e^{-5}$
Scheduler	Cosine Annealing
Weight decay	0.01
Mixed precision	fp16
Epochs	3

F. Evaluation Framework

To evaluate the effectiveness of the guardrail classifier, a structured evaluation framework was implemented across four dimensions:

- Classification reliability
- Category sensitivity
- Inference efficiency
- Calibration robustness

Model performance was measured using standard classification metrics:

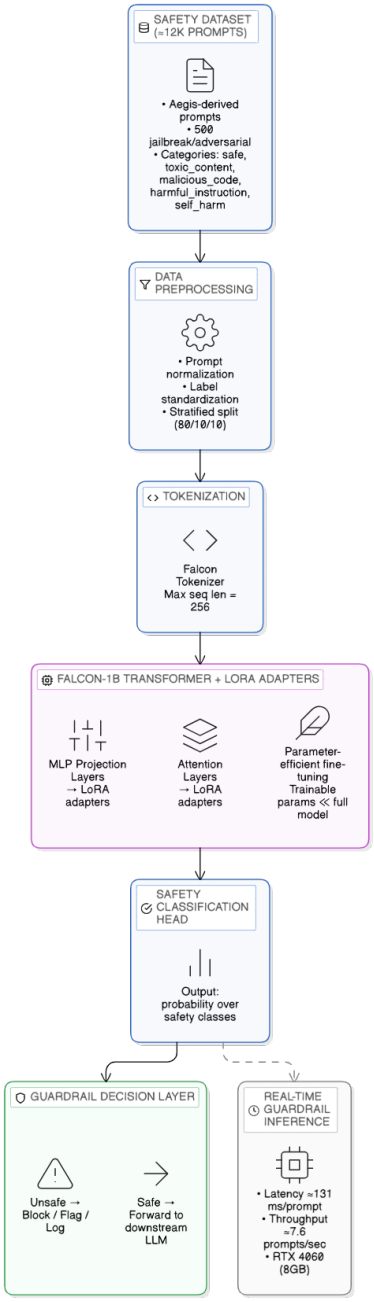

Fig. 1: Proposed LoRA-Based Guardrail Architecture for Safety Classification.

- Accuracy
- Precision
- Recall
- F1 Score
- Matthews Correlation Coefficient
- Cohen's Kappa

Additional diagnostic tools such as confusion matrices and calibration plots were used to analyze model behavior.

IV. Results

This section evaluates the proposed LoRA-based guardrail classifier against existing moderation models using both classification performance and operational efficiency metrics.

A. Model Comparison

The proposed Falcon-1B + LoRA guardrail was compared with several existing moderation models. Table IV summarizes the performance across multiple evaluation metrics.

TABLE IV: Guardrail Model Comparison

Model	Acc	Macro F1	W-F1	Unsafe Rec	FNR	FPR
Granite Guardian 2B	0.865	0.857	0.864	0.929	0.071	0.230
Nemotron 4B	0.909	0.904	0.909	0.946	0.054	0.146
LlamaGuard 1B	0.388	0.150	0.255	∼0.029	High	Low
Falcon-1B + LoRA (Ours)	**0.984**	**0.982**	**0.984**	**∼0.98**	**0.0157**	**Very Low**

The proposed model achieves the highest accuracy and lowest unsafe leakage rate while maintaining a significantly smaller model size compared to competing guardrail systems.

B. Accuracy Comparison

Figure 2 shows the overall classification accuracy across the evaluated guardrail models.

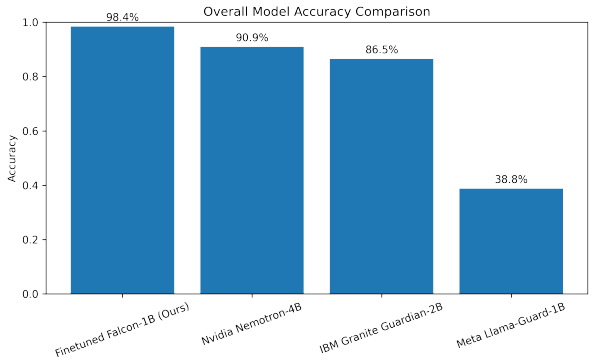

Fig. 2: Overall safety classification accuracy across guardrail models.

The fine-tuned Falcon guardrail achieves an accuracy of 98.4%, outperforming larger baseline models.

C. Unsafe Leakage Analysis

In safety moderation systems, false negatives (unsafe prompts classified as safe) represent a critical risk. Figure 3 compares the unsafe leakage rate (FNR) across models.

The proposed model achieves a leakage rate of 1.57%, significantly lower than competing guardrail models.

D. Reliability Metrics

To evaluate prediction reliability and probability calibration, additional statistical metrics were measured.

TABLE V: Advanced Reliability Metrics

Metric	Value
Matthews Correlation Coefficient	0.9789
Cohen's Kappa	0.9789
Expected Calibration Error	0.0128
Mean Confidence (Correct)	0.998
Mean Confidence (Incorrect)	0.873

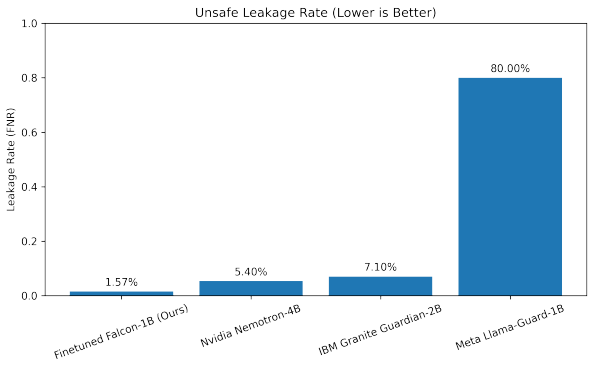

Fig. 3: Unsafe prompt leakage rate (FNR). Lower values indicate safer moderation performance.

These metrics indicate strong agreement between predicted and true labels and well-calibrated confidence estimates.

E. Inference Efficiency

Operational efficiency is critical for real-time moderation systems. Table VI reports the inference performance of the proposed guardrail model.

TABLE VI: Inference Efficiency

Metric	Value
Average Latency	131.9 ms/prompt
Throughput	7.58 prompts/sec
Hardware	RTX 4060 (8GB VRAM)
Batch Size	32
Inference Mode	GPU

The LoRA-adapted Falcon guardrail achieves real-time inference performance on consumer-grade GPUs while maintaining high safety classification accuracy.

REFERENCES

[1] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[2] Z. Ye, D. Li, Z. Hu, T. Lan, J. Sha, S. Zhang, L. Duan, J. Zuo, H. Lu, Y. Zhou, and M. Tang, "mLoRA: Fine-tuning LoRA Adapters via Highly Efficient Pipeline Parallelism," arXiv preprint arXiv:2312.02515, 2023.

[3] Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, and J. Jia, "LongLoRA: Efficient Fine-Tuning of Long-Context Large Language Models," arXiv preprint arXiv:2309.12307, 2023.

[4] Y. Qi et al., "Safety Analysis in the Era of Large Language Models," ScienceDirect, 2025.

[5] C. Akiri, H. Simpson, K. Aryal, A. Khanna, and M. Gupta, "Safety and Security Analysis of Large Language Models: Risk Profile and Harm Potential," arXiv preprint arXiv:2509.10655, 2025.

[6] K. Maity et al., "ToxVidLM: A Multimodal Framework for Toxicity Detection," in *Findings of the Association for Computational Linguistics (ACL)*, 2024.

[7] G. Song et al., "A Study of Multilingual Toxic Text Detection Approaches to Imbalanced Data," *Information*, vol. 12, no. 5, p. 205, 2021.

[8] J. He et al., "Large Language Models for Code: Security Hardening and Vulnerabilities," in *Proc. ACM Conference on Computer and Communications Security (CCS)*, 2023.

[9] N. O. Jaffal, M. Alkhanafseh, and D. Mohaisen, "Large Language Models in Cybersecurity: Applications, Vulnerabilities, and Defense Techniques," *Informatics*, vol. 6, no. 9, p. 216, 2025.

[10] S. Sushma, "Enhanced Toxic Comment Detection Model through Deep Learning," *Future Technology (FUTECH) Journal*, 2025.